WHAT IS CLAIMED IS:

1.   A method of directing messages within a computer system, wherein:

a message is to be directed to a predetermined set of services;

each service executes a command specified by the message;

the message comprises details of the predetermined set of services; and

each service in the predetermined set of services uses said details to determine whether the message should be sent to another service, and if it is determined that the message should be sent to another service transmits the message to an appropriate service.

2.   A method according to claim 1, wherein the message comprises a list of pairs, a first element of each pair representing a service in the predetermined set of services, and a second element of each pair representing a command to be executed by that service.

3.   A method according to claim 1, wherein a plurality of services in the predetermined set of services operate within a single operating system process.

4.   A method according to claim 1, wherein services in the predetermined set of services operate within a plurality of operating system processes.

5.   A method according to claim 1, wherein each service is implemented as a service object which is instance of a respective service class defined in an object oriented programming language.

6.   A method according to claim 5, wherein each service class has an associated service handler class which specifies a method configured to execute a command directed to an instance of the respective service class.

7.   A method according to claim 6, wherein each service object references a plurality of service handler objects which are instances of the respective service handler class.

8.   A method according to claim 1, wherein an attempt is made to locate the appropriate service within the current operating system process.

9.   A method according to claim 8, wherein if the attempt is unsuccessful, the message is transmitted to a messaging service within the current operating system process which is responsible for inter-process communication, and the messaging service transmits the message to a different operating system process.

10.   A method according to claim 1, wherein the message is generated by a composite user interface.

11.   A method according to claim 10, wherein the message is generated by a HTTP request.

12.   A method according to claim 1, wherein the message is directed to a service which generates at least one further message, said further message comprising details of a further set of services to which the further message is to be directed.

13.   A method according to claim 12, wherein each service in the further set of services uses said details of the further set of services to determine whether the further message should be sent to another service, and if it is determined that the message should be sent to another service transmits the message to an appropriate service.

14.   A method according to claim 12, wherein the at least one further message produces a request which is transmitted to at least one source application.

15.   A method according to claim 14, wherein the further message is directed to a service which produces the request in a form determined by predefined configuration data.

16.   A method according to claim 14, wherein data from the source application is used to create a response message in response to said request.

17. A method according to claim 16, wherein said response message comprises details of a set of response services to which the response message is to be directed.

18. A method according to claim 17, wherein each service in said set of response services uses said details of the set of response services to determine whether the response message should be sent to another service, and if it is determined that the message should be sent to another service transmits the message to an appropriate service.

19. A method according to claim 17, wherein the set of response services is the set of further services

20. A method according to claim 17, wherein a service in the set of response services receives a plurality of response messages and combines said plurality of response messages to create a further response message.

21. A method according to claim 20, wherein said further response message comprises details of a set of further response services to which the further response message is to be directed.

22. A method according to claim 21, wherein each service uses said details of the set of further response services to determine whether the further response message should be sent to another service, and if it is determined that the further response message should be sent to another service, transmits the further response message to an appropriate service.

23. A method according to claim 21, wherein said set of further response services is said set of services.

24. A method according to claim 20, wherein the further response message is transmitted to the composite user interface.

25.   A method according to claim 24, wherein the further response message is transmitted to the composite user interface using the HTTP protocol.

26.   A method according to claim 1, wherein said messages are to be directed from a composite user interface to at least one source application.

27.   A method according to claim 26, wherein said at least one source application is a single source application.

28.   A method according to claim 1, wherein:

the message is directed between a composite user interface and at least one source application;

the message is received by an aggregation service which generates at least one further message which produces a request to the at least one source application;

the aggregation service receives a plurality of response messages containing data generated by the at least one source application; and

the aggregation service uses said plurality of response messages to generate a further response message which is transmitted to the composite user interface.

29.   A method according to claim 28, wherein the or each further message comprises details of a further set of services to which the further message is to be directed, and each service uses said details of said further set of services to determine whether the further message should be sent to another service, and if it is determined that the message should be sent to another service transmits the message to an appropriate service.

30.   A method according to claim 28, wherein each response message comprises details of a respective set of response services to which the response message is to be directed, and each service uses said details of a respective set of response services to determine whether the response message should be sent to another service, and if it is determined that the response message should be sent to another service transmits the message to an appropriate service;

31.   A method according to claim 28, wherein the further response message comprises a further set of response services to which the further response message is to be directed, and each service uses said details of the further set of response services to determine whether the further response message should be sent to another service, and if it is determined that the further response message should be sent to another service transmits the further response message to an appropriate service.

32.   A method according to claim 28, wherein said data contained in the plurality of response messages contains user interface elements.

33.   A method according to claim 32, wherein said aggregation service combines said user interface elements to generate said composite user interface in accordance with predefined configuration data.

34.   A method according to claim 33, wherein said aggregation service generates additional user interface elements which are combined with said user interface elements to generate said composite user interface.

35.   A method according to claim 33, wherein said configuration data is arranged in a hierarchical data structure.

36.   A method according to claim 35, wherein a first entity within the hierarchical data structure represents the composite user interface, and child entities of said first entity represent the plurality of user interface elements for inclusion in the composite interface.

37.   A method according to claim 36, comprising creating a run-time data structure having a structure substantially corresponding to that of said hierarchical data structure.

38.   A method according to claim 37, wherein at least some entities of said run-time data structure contain state information.

39.   A method according to claim 38, comprising creating an object data structure comprising a plurality of objects, the object data structure having a structure substantially corresponding to at least part of said hierarchical data structure, and each object representing a received user interface element.

40.   A method according to claim 39, comprising receiving a user request at a first entity within said run-time data structure corresponding to said first entity within the hierarchical data structure.

41.   A method according to claim 40, wherein said user request comprises at least one parameter, and said at least one parameter is stored at said first entity.

42.   A method according to claim 40, comprising traversing said run-time data structure to execute at least one action associated with a child entity of said first entity within said run-time data structure.

43.   A method according to claim 41, wherein said at least one action comprises generating a request for a user interface element represented by said child entity.

44.   A method according to claim 43, wherein said request comprises state information from at least one entity within said run-time data structure.

45.   A method according to claim 36, wherein each child entity has an associated parameter indicating whether the respective source user interface element is mandatory.

46.   A method according to claim 45, wherein the aggregation service:
        receives a user interface element;
        stores data indicative of receipt of said user interface element in a data structure associated with said hierarchical data structure; and
        when all source user interface elements having an associated parameter indicating that the source user interface is mandatory have been received, combines said plurality of source data items to generate said composite user interface.

47.   A method according to claim 46, wherein the composite user interface is generated in an internal format.

48.   A method according to claim 47, wherein said composite user interface represented in said internal format is converted to an output format using output configuration data.

49.   A method according to claim 28, wherein a transformation service receives data from the at least one source application and transforms said data into an internal format, said transformed data being contained in said plurality of response messages.

50.   A method according to claim 49, wherein the transformation service applies regular expressions to transform data.

51.   A method according to claim 49, wherein the transformation service applies transformations defined within a class defined in an object oriented programming language to transform data.

52.   A method according to claim 49, wherein the data received from the at least one source application is a HTML page.

53.   A method according to claim 52, wherein the transformation service is configured to recognise a plurality of HTML pages, and is configured to extract predetermined user interface elements from recognised HTML pages.

54.   A method according to claim 52, wherein the transformation service processes HTML pages to amend references to other HTML pages in accordance with predefined configuration data.

55.   A method according to claim 49, wherein the transformation service transforms data using configuration data contained in a hierarchical data structure, said hierarchical data structure containing an entity for each source application.

56.   A method according to claim 28, wherein the aggregation service is configured to expect to receive a predetermined number of response messages in response to transmission of said further message, and said further response message is generated when the predetermined number of response messages has been received.

57.   A method according to claim 28, wherein the aggregation service is configured to expect to receive a predetermined number of mandatory response messages, and predetermined number of optional response messages, and said further message is generated when the predetermined number of mandatory response messages has been received.

58.   A system for directing messages within a computer system, wherein:
        the system comprises a plurality of services;
        a message is to be directed to a predetermined set of services; and
        the message comprises details of the predetermined set of services;
each service in the predetermined set of service comprising:
        executing means for executing a command specified by the message
        determining means for using said details to determine whether the message should be sent to another service; and
        transmitting means for transmitting the message to an appropriate service if it is determined that the message should be sent to another service.

59.   A system according to claim 58, wherein the message comprises a list of pairs, a first element of each pair representing a service in the predetermined set of services, and a second element of each pair representing a command to be executed by that service.

60.   A system according to claim 58, wherein a plurality of services in the predetermined set of services operate within a single operating system process.

61.   A system according to claim 58, wherein services in the predetermined set of services operate within a plurality of operating system processes.

62.   A system according to claim 58, wherein each service is implemented as an object which is instance of a respective service class defined in an object oriented programming language.

63.   A system to claim 62, wherein each service class has an associated service handler class which specifies a method configured to execute a command directed to an instance of the respective service class.

64.   A method according to claim 63, wherein each service object references a plurality of service handler objects which are instances of the respective service handler class.

65.   A system according to claim 58, wherein the transmitting means of each service attempts to locate the appropriate service within the current operating system process.

66.   A system according to claim 65, wherein if the attempt is unsuccessful, the transmitting means transmits the message to a messaging service within the current operating system process which is responsible for inter-process communication, and the messaging service transmits the message to a different operating system process.

67.   A system according to claim 58, wherein the message is generated by a composite user interface.

68.   A system according to claim 67, wherein the message is generated by a HTTP request.

69.   A system according to claim 58, comprising means to direct the message to a service which generates at least one further message, said further message comprising details of a further set of services to which the further message is to be directed.

70.   A system according to claim 69, wherein each service in the further set of services comprises determining means which uses said details of the further set of services to determine whether the further message should be sent to another service, and

86

transmitting means for transmitting the message to an appropriate service if it is determined that the message should be sent to another service.

71.   A system according to claim 69, comprising means to produce a request which is transmitted to the at least one source application from the at least one further message

72.   A system according to claim 71, comprising means to direct the further message to a service which produces the request in accordance with predefined configuration data.

73.   A system according to claim 72, comprising means for creating a response message in response to said request using data from the at least one source application

74.   A system according to claim 73, wherein said response message comprises details of a set of response services to which the response message is to be directed.

75.   A system according to claim 74, wherein each service in said set of response services comprises determining means for determining whether the response message should be sent to another service from said details of the set of response services, and transmitting means for transmitting the message to an appropriate service if it is determined that the message should be sent to another service.

76.   A system according to claim 74, wherein the set of response services is the set of further services.

77.   A system according to claim 74, wherein a service in the set of response comprises means for receiving a plurality of response messages, and means for combining said plurality of response messages to create a further response message.

78.   A system according to claim 77, wherein said further response message comprises a set of further response services to which the further response message is to be directed.

87

79.   A system according to claim 78, wherein each service comprises determining means for determining whether the further response message should be sent to another service using said details of the set of further response services, and transmitting means for transmitting the further response message to an appropriate service, if it is determined that the further response message should be sent to another service.

80.   A system according to claim 78, wherein said set of further response services is said set of services.

81   A system according to claim 77, comprising means for transmitting the further response message to the composite user interface.

82.   A system according to claim 81, comprising means for transmitting the further response message to the composite user interface using the HTTP protocol.

83.   A system according to claim 58, wherein said at least one source application is a plurality of source applications.

84.   A system according to claim 58, wherein said at least one source application is a single source application.

85.   A system according to claim 58, wherein:
        the system comprises a composite user interface and at least one source application;
        the system comprises a plurality of services, including an aggregation service;
        a message is to be directed to a predetermined set of services; and
        the message comprises details of the predetermined set of services;
each service in the predetermined set of service comprising:
        executing means for executing a command specified by the message
        determining means for using said details to determine whether the message should be sent to another service; and
        transmitting means for transmitting the message to an appropriate service and if it is determined that the message should be sent to another service; and

the aggregation service comprising:

means for generating at least one further message;

means for producing a request to the at least one source application from said further message;

means for receiving a plurality of response messages containing data generated by the at least one source application;

means for generating a further response message from said plurality of response messages; and

means for transmitting said further response message to the composite user interface.

86.   A system according to claim 85, wherein the or each further message comprises details of a further set of services to which the further message is to be directed, and each service comprises determining means for using said details of said further set of services to determine whether the further message should be sent to another service, and transmitting means for transmitting said further message to an appropriate service if it is determined that the further message should be sent to another service.

87.   A system according to claim 85, wherein each response message comprises a details of a respective set of response services to which the response message is to be directed, and each service comprises determining means for using said details of a respective set of response services to determine whether the response message should be sent to another service, and transmitting means for transmitting said response message to an appropriate service if it is determined that the response message should be sent to another service.

88.   A system according to claim 85, wherein the further response message comprises a further set of response services to which the further response message is to be directed, and each service determining comprises means for using said details of the further set of response services to determine whether the further response message should be sent to another service, and transmitting means for transmitting said further response message to an appropriate service if it is determined that the message should be sent to another service.

89.   A system according to claim 85, wherein said data contained in the plurality of response messages contains user interface elements.

90.   A system according to claim 89, wherein said aggregation service comprises combining means for combining said user interface elements to generate said composite user interface in accordance with predefined configuration data.

91.   A system according to claim 90, wherein said aggregation service comprises means for generating additional user interface elements which are combined with said user interface elements by said combining means to generate said composite user interface.

92.   A system according to claim 90, wherein said configuration data is arranged in a hierarchical data structure.

93.   A system according to claim 92, wherein a first entity within the hierarchical data structure represents the composite user interface, and child entities of said first entity represent the plurality of user interface elements for inclusion in the composite interface.

94.   A system according to claim 93, wherein each child entity has an associated parameter indicating whether the respective source user interface element is mandatory.

95.   A system according to claim 94, wherein the aggregation service comprises:
       means for receiving a user interface element;
       means for storing data indicative of receipt of said user interface element in a data structure associated with said hierarchical data structure; and
       means combining said plurality of source data items to generate said composite user interface when all source user interface elements having an associated parameter have been received.

96.   A system according to claim 95, comprising means for generating the composite user interface in an internal format.

97.   A system according to claim 96, comprising means for converting said composite user interface represented in said internal format into an output format using output configuration data.

98.   A system according to claim 58, comprising a transformation service, said transformation service comprising:

　　　　means for receiving data from the at least one source application; and

　　　　means for transforming said data into an internal format which is containing within said plurality of response messages.

99.   A system according to claim 98, wherein the transformation service comprises means for applying regular expressions to transform data.

100.   A system according to claim 98, wherein the transformation service comprises means for applying transformations defined within a class defined in an object oriented programming language to transform data.

101.   A system according to claim 98, wherein the data received from the at least one source application is a HTML page.

102.   A system according to claim 101, wherein the transformation service comprises means to recognise a plurality of HTML pages, and means to extract predetermined user interface elements from recognised HTML pages.

103.   A system according to claim 101, wherein the transformation service comprises means to process HTML pages to amend references to other HTML pages in accordance with predefined configuration data.

104.   A system according to claim 98, wherein the transformation service comprises means to transform data using configuration data contained in a hierarchical data

structure, said hierarchical data structure containing an entity for each source application

105. A system according to claim 85, wherein the aggregation service is configured to expect to receive a predetermined number of response messages in response to transmission of said further message, and the aggregation service comprises means to generate said further response message when the predetermined number of response messages has been received.

106. A system according to claim 85, wherein the aggregation service is configured to expect to receive a predetermined number of mandatory response messages, and predetermined number of optional response messages, and said aggregation service comprises means to generate said further message when the predetermined number of mandatory response messages has been received.

107. A carrier medium carrying computer readable program code means for controlling a computer to carry out the method of claim 1.

108. A computer apparatus for directing messages between a composite user interface and at least one source application, the apparatus comprising:

a program memory containing processor readable instructions; and

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of claim 1.

109. A method for generating a composite user interface for presentation to a user, the method comprising:

generating requests for a plurality data items for inclusion in the interface, transmitting each request to one of a plurality of source applications; and

combining data items received in response to at least one of said requests to generate the user interface;

wherein at least some of the predetermined plurality of data items are mandatory, and at least some of the predetermined plurality of data items are optional, and the composite user interface is generated when all mandatory data items have been received.

110. A method according to claim 109, comprising generating said plurality of requests from a single request.

111. A method according to claim 110, wherein said single request is a HTTP request entered by a user using a web browser.

112. A method according to claim 111, wherein said single request is received by an aggregation service, and the aggregation service generates the plurality of requests in accordance with predetermined configuration data.

113. A method according to claim 112, wherein said configuration data is arranged within a hierarchical data structure.

114. A method according to claim 113, wherein a first entity within the hierarchical data structure represents the composite user interface, and child entities of said first entity represent the plurality of data items for inclusion in the composite interface.

115. A method according to claim 114, wherein each child entity has an associated parameter indicating whether the respective data item is mandatory.

116. A method according to claim 115, wherein the aggregation service:
    locates said first entity in the hierarchical data structure;
    locates all child entities of said first entity;
    generates a request for each data item represented by a child entity; and
    transmits each request to a respective source application;

117. A method according to claim 116, wherein the aggregation service additionally,
    receives a requested data item;

stores data indicative of receipt of said requested data item in a data structure associated with said hierarchical data structure; and

when all data items having associated parameter indicating that the data item is mandatory have been received, combining said plurality of data items to generate said composite user interface.

118.   A method according to claim 117, wherein the composite user interface is generated in an internal format.

119.   A method according to claim 118, wherein said composite user interface represented in said internal format is converted to an output format using output configuration data.

120.   A system for generating a composite user interface for presentation to a user, the system comprising:

        means for generating requests for a plurality data items for inclusion in the interface;

        means for transmitting each request to one of a plurality of source applications;

        means for combining data items received in response to at least one of said requests to generate the user interface;

        wherein at least some of the predetermined plurality of data items are mandatory, and at least some of the predetermined plurality of data items are optional, and the composite user interface is generated when all mandatory data items have been received.

121.   A system according to claim 120, further comprising means for generating said plurality of requests from a single request.

122.   A system according to claim 121, wherein said single request is a HTTP request entered by a user using a web browser.

123.   A system according to claim 122, comprising an aggregation service, said aggregation service comprising:

means for receiving said single request; and

means for generating the plurality of requests in accordance with predetermined configuration data.

124. A system according to claim 123, wherein said configuration data is arranged within a hierarchical data structure.

125. A system according to claim 124, wherein a first entity within the hierarchical data structure represents the composite user interface, and child entities of said first entity represent the plurality of data items for inclusion in the composite interface.

126. A method according to claim 125, wherein each child entity has an associated parameter indicating whether the respective data item is mandatory.

127. A system according to claim 126, wherein the aggregation service comprises:

means for locating said first entity in the hierarchical data structure;

means for locating all child entities of said first entity;

means for generating a request for each data item represented by a child entity; and

means for transmitting each request to a respective source application;

128. A system according to claim 127, wherein the aggregation service further comprises,

means for receiving a requested data item;

means for storing data indicative of receipt of said requested data item in a data structure associated with said hierarchical data structure; and

means for combining said plurality of data items to generate said composite user interface when all data items having associated parameter indicating that the data item is mandatory have been received.

129. A system according to claim 128, wherein the composite user interface is generated in an internal format.

130.   A system according to claim 129, wherein said composite user interface represented in said internal format is converted to an output format using output configuration data.

131.   A carrier medium carrying computer readable program code means for controlling a computer to carry out the method claim 109.

132.   A computer apparatus for generating a composite user interfacecomprising:

a program memory containing processor readable instructions; and

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of claim 109.

133.   A method for generating a composite user interface for presentation to a user, said composite user interface comprising a plurality of user interface elements generated from source interface elements provided by at least one source application, the method comprising:

generating a plurality of request messages;

transmitting each request message to an appropriate source application;

receiving a plurality of source interface elements from the at least one source application;

comparing each received source interface element with a plurality of predefined source interface templates; and

if said received source interface element matches a predefined source interface template, extracting at least one user interface element for inclusion in said composite user interface.

134.   A method according to claim 133, wherein at least one of said source interface elements is a HTML document.

135.   A method according to claim 133, further comprising:

creating an internal representation of each extracted user interface element;

forwarding said internal representations to an aggregation service;

combining said internal representations to create an internal representation of the composite user interface.

136.   A method according to claim 134, wherein said combination is effected in accordance with predefined configuration data.

137.   A method according to claim 135, further comprising transforming said internal representation of the composite user interface into an output format specified by predefined configuration data.

138.   A system for generating a composite user interface for presentation to a user, said composite user interface comprising a plurality of user interface elements generated from source interface elements provided by at least one source application, the system comprising:

means for generating a plurality of request messages;

means for transmitting each request message to an appropriate source application;

means for receiving a plurality of source interface elements from the at least one source application;

means comparing each received source interface element with a plurality of predefined source interface templates; and

means for comparing said received source interface element with a plurality of predefined source interface templates; and

means for extracting the at least one user interface element for inclusion in said composite user interface if said comparing means determines a match.

139.   A carrier medium carrying computer readable program code means for controlling a computer to carry out the method of claim 133.

140.   A computer apparatus for generating a composite user interface comprising:

a program memory containing processor readable instructions; and

97

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of claim 133.

141.   A method of processing messages within a computer system, comprising:

receiving a message comprising details of a predetermined set of processing modules, and a command;

executing said command; and

transmitting the message to a processing module in said predetermined set of processing modules.

142.   A method of directing messages between a composite user interface and at least one source application, comprising:

receiving a message generated by said composite user interface;

carrying out processing specified by said message;

generating at least one further message to be directed to the at least one source application, said at least one further message comprising details of a set of processing modules to which the further message is to be directed; and

directing said at least one further message to a processing module in said set of processing modules.

143.   A method of aggregating user interface data received from a plurality of source applications comprising:

requesting a set of user interface elements;

receiving a plurality of user interface elements; and

aggregating received user interface elements to form a composite user interface when a predetermined subset of said set of user interface elements has been received.

144.   A computer apparatus configured for processing a message comprising details of a set of processing modules to which the message is to be directed, and at least one command, the apparatus comprising:

a code module adapted to receive the message;

a code module adapted to execute said at least one command;

a code module adapted to determine a processing module within said set processing modules to which the message should be transmitted; and

a code module adapted to transmit said message to said determined processing module.

145.   A method of directing messages between a composite user interface and at least one source application, comprising:

receiving a message generated by said composite user interface, said message comprising a predetermined set of processing modules to which the message should be directed;

carrying out processing specified by said message; and

directing said message to a processing module in said set of processing modules.

146.   A method for generating a composite user interface for communication with a plurality of source applications, the method comprising:

receiving a message generated by said composite user interface;

identifying from data within said message a first entity within a hierarchical data structure;

identifying at least one child entity of said first entity within said hierarchical data structure; and

carrying out processing specified by data stored at the or each child entity, the processing comprising generating a further message for transmission to at least one of said plurality of source applications.